Prof. Dr. Karsten Urban
Dipl.-Math. oec. Klaus Stolle
Institut für Numerische Mathematik
Universität Ulm

SoSe 2014
Blatt 1
25.04.2014

Angewandte Numerik 1

Abgabetermin: Freitag, 16.05.2014, **vor der Übung**

Dieses Übungsblatt hat eine Bearbeitungszeit von drei Wochen. Nicht alle Aufgaben können mit dem Stoff der Vorlesung vom 24.04.2014 bearbeitet werden.
Punkte, die mit einem * gekennzeichnet sind, sind Zusatzpunkte.

Aufgabe 0 *(Organisatorisches)* (2 Punkte)

a) Bitte melden Sie sich für die Mailingliste an.

b) Bitte melden Sie sich im `slc` zur Vorlesung an.

c) Bitte tragen Sie sich spätestens bis **Samstag, 26.04.2014, 24:00 Uhr** im `slc` für ein Tutorium ein, sofern Sie eines besuchen wollen.

d) Abgabe der Übungsblätter nur **zu zweit**.

Aufgabe 1 *(Basiswechsel)* (6* Punkte)

Für die Anwendung von Bedeutung sind bei der Zahldarstellung die Basen $b = 10$ (Dezimalzahlen), $b = 2$ (Binärzahlen), $b = 8$ (Oktalzahlen) und $b = 16$ (Hexadezimalzahlen).
Bei Hexadezimalzahlen treten Ziffern mit Werten zwischen 1 und 15 auf. Damit alle Ziffern einstellig notiert werden können, werden für die Ziffern mit den Werten 10 bis 15 die Buchstaben A bis F verwendet.
Ergänzen Sie die folgende Tabelle (geben Sie dabei alle Rechnungen an):

Dezimal	Dual	Oktal	Hexadezimal
30.125			
	11011.011		
		75.21	
			5.C

Aufgabe 2 *(Programmieraufgabe: Zahlensysteme)* (6 Punkte)

Wir beschränken uns zunächst auf natürliche Zahlen $n \in \mathbb{N}$ und suchen deren Zahlendarstellung zu einer gegeben Basis $b \in \mathbb{N}_{\geq 2} := \{2, 3, 4, ...\}$. Gesucht ist also folgende Darstellung

$$n = \sum_{k=0}^{m} a_k b^k$$

mit den Koeffizienten $a_k \in \{0, 1, ..., b-1\}$ für $k = 0, ..., m$.

a) Schreiben Sie eine Matlab-Funktion

```
a = convert2basis(b, n)
```

mit Übergabeparameter $b \in \mathbb{N}_{\geq 2}$ und $n \in \mathbb{N}$, welche die Koeffizienten der Zahldarstellung bzgl. der Basis b zurück gibt, also

$$a = (a_m, a_{m-1}, ..., a_1, a_0).$$

b) Testen Sie die Funktion mit Hilfe des Skriptes `test_convert2basis.m`, welches auf der Homepage verfügbar ist.

Aufgabe 3 *(Programmieraufgabe: Berechnung des Wertes einer Gleitpunkt-Darstellung und umgekehrt)* (8+8* Punkte)

Sei $b \in \mathbb{N}_{\geq 2} := \{2, 3, 4, ...\}$ und $x \in [b^{-b^n}, b^{b^n - 1})$ in der (eindeutigen) Darstellung

$$x = b^{\ell} \cdot f = b^{t \cdot \ell(\mathbf{v})} \cdot \sum_{j=1}^{\infty} d_j b^{-j} \qquad \text{mit} \qquad \ell(\mathbf{v}) = \sum_{j=0}^{n-1} v_j b^j,$$

wobei wir annehmen, dass $d_j < b - 1$ für unendlich viele j.

a) Schreiben Sie eine Matlab-Funktion

```
x = value(b, d, v, t)
```

die für eine gegebene Gleitpunktdarstellung, d.h. für folgende Eingabeparameter

- eine Basis `b` $\in \mathbb{N}_{\geq 2}$
- einen Zeilenvektor `d` $= (d_1, d_2, ..., d_m)$
- einen Zeilenvektor `v` $= (v_{n-1}, ...v_1, v_0)$
- `t` $\in \{-1, 1\}$.

den Wert `x` der Gleitpunkt-Darstellung berechnet.

b) Testen Sie die Funktion mit Hilfe des Skriptes `test_value.m`, welches auf der Homepage verfügbar ist.

c) Schreiben Sie eine Matlab-Funktion (mit Hilfe von Algorithmus 1)

```
[d, v, t] = flp(b, m, n, x)
```

welche für die Eingabeparameter

- `b`: eine Basis $b \in \mathbb{N}_{\geq 2}$
- `m`: die Mantissenlänge
- `n`: die Exponetenlänge
- `x`: die zu konvertierende Zahl

die Gleitpunkt-Darstellung (ohne Runden) mit $d_1 \neq 0$ von `x` berechnet und `d`, `v` und `t` zurückliefert. Ein Aufruf sieht beispielsweise wie folgt aus:

```
[d, v, t] = flp(2, 3, 3, 0.0625)
d = [1, 0, 0]
v = [0, 1, 1]
t = -1
```

Algorithm 1 Computation of $\mathbf{d} = (d_1, \ldots, d_m)$

if $0 < bx < 1$ **then**
    Find the smallest $k \in \mathbb{N}$ such that $y_1 := b^{k+1}x \geq 1$.
    Set $\ell = \ell(\mathbf{v}) = k$ and $t = -1$.
    for $j = 1, \ldots, m$ **do**
        Compute the smallest $d_j \in \{0, \ldots, b-1\}$ such that $y_j - d_j < 1$.
        Set $y_{j+1} = b(y_j - d_j)$.
    end for
else if $bx \geq 1$ **then**
    Find the smallest $k \in \mathbb{N}_0$ such that $x < b^k$
    Set $\ell = \ell(\mathbf{v}) = k$ and $t = 1$.
    Set $y_1 := bx$.
    for $j = 1, \ldots, m$ **do**
        Compute the smallest $d_j \in \{0, \ldots, b-1\}$ such that $y_j - d_j b^k < b^k$.
        Set $y_{j+1} = b(y_j - d_j b^k)$.
    end for
end if

d) Testen Sie die Funktion mit Hilfe des Skriptes `test_flp.m`, welches auf der Homepage verfügbar ist.

Aufgabe 4 *(Maschinenzahlen)* (4+4 Punkte)

a) Bestimmen Sie für $a = \frac{3}{5}$ und $b = \frac{4}{7}$ die Darstellungen in $\tilde{a} \in \mathbb{M}(2,5,3)$ und $\tilde{b} \in \mathbb{M}(2,3,3)$ (mit Standardrundung).

b) Berechnen Sie

$$x_1 := \tilde{a} \ominus \tilde{b} \quad \text{in } \mathbb{M}(2,5,3) \qquad \text{und} \qquad x_2 := \tilde{a} \ominus \tilde{b} \quad \text{in } \mathbb{M}(2,3,3)$$

sowie den jeweiligen relativen Fehler. Was fällt auf?

Aufgabe 5 *(Kondition des Problems des Schnittpunkts von Geraden)* (2+2+4 Punkte)

Gegeben seien zwei Geraden G_1, G_2 im $\mathbb{R}^2$.

$$G_1 = \{(x_1, x_2) \in \mathbb{R}^2;\ a_{1,1}x_1 + a_{1,2}x_2 = b_1\}$$
$$G_2 = \{(x_1, x_2) \in \mathbb{R}^2;\ a_{2,1}x_1 + a_{2,2}x_2 = b_2\}$$

a) Berechnen Sie den Schnittpunkt $x = (x_1, x_2)^T$ der beiden Geraden in Abhängigkeit der Koeffizienten $a_{i,j}, i, j = 1, 2$ und $b = (b_1, b_2)^T$. Schreiben Sie dazu das Problem als lineares Gleichungssystem und gehen Sie davon aus, dass die Matrix $A := (a_{i,j})_{i,j=1,2}$ regulär ist.

b) Gehen Sie davon aus, dass es eine Störung in den Eingabedaten b gibt. Geben Sie jeweils eine möglichst einfache Kombination der Werte für a_{ij} und b an, so dass sich ein gut bzw. ein schlecht konditioniertes Problem ergibt. Skizzieren Sie den Sachverhalt, der sich mit den von Ihnen gewählten Werten ergibt.

c) Gehen Sie wiederum davon aus, dass es eine Störung in den Eingabedaten b gibt. Das Problem $x = \varphi(A, b) = A^{-1}b$ sei also von der Form $\varphi(b) = \varphi(A, b)$, $\varphi : X \to Y$ mit $X = Y = \mathbb{R}^2$. Berechnen Sie

die absolute Kondition

$$\kappa_{\varphi,abs} = \frac{\|\Delta y\|_Y}{\|\Delta x\|_X}$$

des Schnittpunkt-Problems. Verwenden Sie hierbei die Maximumsnorm $\|x\|_\infty := \max_{i=1,\dots,n} |x_i|$.

Aufgabe 6 *(Lösen quadratischer Gleichungen, Auslöschung vermeiden)* (3+2+2+4 Punkte)

Für $p, q \in \mathbb{R}$, $p^2 >> q$ sollen die Lösungen $x_1 \leq x_2$ der quadratischen Gleichung

$$x^2 - 2px + q = 0$$

berechnet werden. Hierbei soll Auslöschung (d.h. Subtraktion nahezu gleich großer Zahlen) vermieden werden.

a) Begründen Sie, dass die Gleichung zwei reelle Lösungen hat und dass man bei Verwendung der üblichen Formel $x_{1/2} = p \pm \sqrt{p^2 - q}$ auf Auslöschung achten muss. Unter welchen Bedingungen tritt Auslöschung auf?

b) Zeigen Sie $x_1 x_2 = q$ und $x_1 + x_2 = 2p$.

c) Schreiben Sie eine Matlab-Funktion

```
[x1, x2] = nullstellen(p, q),
```

die x_1 und x_2 ohne Auslöschung berechnet.

d) Schreiben Sie ein Matlab-Skript, das Ihre Funktion aus Aufgabenteil c) für $q = 1$ und $p = \pm 100^k, k = 2, 3, 4$ testet. Vergleichen Sie dabei die Ergebnisse Ihrer Funktion `nullstellen(p,q)` mit denen, die die auslöschungsbehaftete Formel aus Aufgabenteil a) liefert.

Tipp: Schauen Sie sich hierzu das Beispiel 2.6.4 aus der Vorlesung an.

Aufgabe 7 *(Vektornormen)* (4+1 Punkte)

Wir betrachten den Vektorraum $X := \mathbb{R}^2$.

a) Zeigen Sie, dass durch

$$\|x\|_1 := |x_1| + |x_2|, \qquad \|x\|_2 := \sqrt{|x_1|^2 + |x_2|^2},$$

Normen auf X definiert sind.

Hinweis: Um für die Euklidische Norm die Dreiechsungleichung $\|x + y\|_2 \leq \|x\|_2 + \|y\|_2$ zu zeigen, schätzen Sie $\|x+y\|_2^2$ geeignet ab. Verwenden Sie hierbei (ohne Beweis) die so genannte Cauchy-Schwarz Ungleichung:

$$|x^T y| \leq \|x\|_2 \|y\|_2.$$

b) Skizzieren Sie für die Normen $\|\cdot\|_*$ in Aufgabenteil a) und die Norm $\|x\|_\infty := \max\{|x_1|, |x_2|\}$ jeweils die Menge $\{x \in \mathbb{R}^2;\ \|x\|_* = 1\}$.

Hinweise:
Die Programmieraufgaben sind in Matlab zu erstellen. Senden Sie alle Files in einer E-mail mit dem Betreff **Loesung-Blatt01** an **angewandte.numerik@uni-ulm.de** (Abgabetermin jeweils wie beim Theorieteil). Drucken Sie zusätzlich allen Programmcode sowie die Ergebnisse aus und geben Sie diese vor der Übung ab. Der Source Code sollte strukturiert und, wenn nötig, dokumentiert sein.